Scientia Iranica B (2012) 19 (3), 503–512

Sharif University of Technology

Scientia Iranica

Transactions B: Mechanical Engineering

www.sciencedirect.com

Research note

3D implicit mesh-less method for compressible flow calculations

S. Sattarzadeh, A. Jahangirian *

Department of Aerospace Engineering, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, P.O. Box 15875-34413, Iran

Received 15 February 2011; revised 8 February 2012; accepted 9 April 2012

KEYWORDS

Mesh-less method;
Implicit method;
Compressible flow;
3D geometries.

Abstract An implicit mesh-less method is developed for calculation of compressible flows around three dimensional complex geometries. The algorithm is applied directly to the differential form of the governing equations using least-square formulation. A dual-time implicit time discretization scheme is developed, and the computational efficiency is enhanced by adopting accelerating techniques, such as local time stepping, residual smoothing and enthalpy damping. Two different artificial dissipation techniques are employed for stability preservation and it is shown that the scalar one is more efficient in terms of accuracy and computational time. The capabilities of the method are demonstrated by flow computations around different geometries under subsonic and transonic flow conditions. Results are presented which indicate good agreement with experimental and other reliable numerical data. The method is shown to reduce computational time by about 50% compared with the alternative explicit method.

1. Introduction

The increased power of computers during the last few years has made the computation of flows over complex geometries more viable. One major problem is the ability to generate high quality mesh. This is really important, particularly where more sophisticated geometrics are concerned. Alternatively, mesh-less schemes are presented, in which point connections are not necessary. Mesh-less methods are based on points, either local or global, which depend on partial differential equations [1]. Some of the earliest research includes the work of Batina [2] who used the least squares method to compute inviscid and viscous flow equations. The Finite Point Method (FPM) was developed by Onate et al. [3] using a polynomial basis. Deshpande and others used least squares methods in the context of the kinetic upwind method (LSKUM) [4]. An upwind scheme based on least squares was developed by Sridar and Balakirshnan [5]. Katz and Jameson presented a mesh-less method, based on the CUSP (Convective Upwind and Split Pressure) scheme, to compute inviscid [6] and viscous [7] flows in two dimensions. In Galerkin-type methods, a background grid is required for numerical integration purposes [8].

Higher-order mesh-less methods are presented not only to minimize the lack of conservation, which is a drawback of mesh-less methods, but also to improve solution accuracy. Another problem with mesh-less methods is the computational cost. At any iteration, the computational cost of mesh-less methods is higher than mesh based ones [1,9]. This problem becomes more crucial when 3D geometries are of concern. A normal solution to this problem is using accelerating techniques, such as the multi-cloud approach [10].

In the present work, a dual-time implicit mesh-less method is developed for three dimensional geometries, which is shown to perform well in terms of accuracy and computational cost compared with explicit and other alternative mesh-less methods.

2. Governing flow equations

Three-dimensional Euler equations consisting of mass, momentum and energy conservation laws that govern the motion of an inviscid flow are:

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial K}{\partial z} = 0. \quad (1)$$

Here:

* Corresponding author.
E-mail address: ajahan@aut.ac.ir (A. Jahangirian).

Peer review under responsibility of Sharif University of Technology.

Production and hosting by Elsevier

1026-3098 © 2012 Sharif University of Technology. Production and hosting by Elsevier B.V. Open access under CC BY license.
doi:10.1016/j.scient.2012.04.013

$$W = \begin{pmatrix} \rho u \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \qquad F = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho v u \\ \rho w u \\ \rho E u + P u \end{pmatrix},$$
$$G = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ \rho w v \\ \rho E v + P v \end{pmatrix}, \qquad K = \begin{pmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + P \\ \rho E w + P w \end{pmatrix} \tag{2}$$

where P, ρ, u, v, w and E indicate the pressure, density, Cartesian velocity components and total energy, respectively. For a perfect gas, the following relation can also be used:

$$P = (\gamma - 1)\left[\rho E - \frac{\rho(u^2 + v^2 + w^2)}{2}\right]. \tag{3}$$

Here, γ is the ratio of specific heat.

2.1. Spatial discretization

In this research, the flow equations are solved in the conservation form. The mesh-less algorithm is applied directly to the differential form of the governing equations. According to most mesh-less methods, a least-square formulation is utilized [9,11]. Suppose that C_i is the set of cloud points for a given point, i (Figure 1). φ_{ij} is the value of any function, φ, at the mid-point of the edge, ij, where j is in the cloud of point i [6]. Assuming this function varies linearly along the edge, ij, and using Taylor's formula about i to any of its cloud points, the following is obtained:

$$\left(\frac{\partial \phi}{\partial x}\right)_i \Delta x_{ij} + \left(\frac{\partial \phi}{\partial y}\right)_i \Delta y_{ij} + \left(\frac{\partial \phi}{\partial z}\right)_i \Delta z_{ij} = \Delta \phi_{ij}$$
$$\Delta x_{ij} = x_j - x_i \qquad \Delta y_{ij} = y_j - y_i$$
$$\Delta z_{ij} = z_j - z_i, \qquad \Delta \phi_{ij} = \phi_j - \phi_i. \tag{4}$$

Similar equations could be written for all cloud points associated with point i, subject to an arbitrary weighting factor, ω_i. This leads to the following non-square matrix:

$$\begin{pmatrix} \omega_{i1}\Delta x_{i1} & \omega_{i1}\Delta y_{i1} & \omega_{i1}\Delta z_{i1} \\ \cdots & \cdots & \cdots \\ \omega_{im}\Delta x_{im} & \omega_{im}\Delta y_{im} & \omega_{im}\Delta z_{im} \end{pmatrix} \begin{bmatrix} \left.\frac{\partial \phi}{\partial x}\right|_i \\ \left.\frac{\partial \phi}{\partial y}\right|_i \\ \left.\frac{\partial \phi}{\partial z}\right|_i \end{bmatrix} = \begin{bmatrix} \omega_{i1}\Delta \phi_{i1} \\ \cdots \\ \omega_{im}\Delta \phi_{im} \end{bmatrix} \tag{5}$$
$$\omega_{ij} = \frac{1}{d_{ij}^q}.$$

Here, d_{ij} is the distance between point i and its neighbor, j. In this work, a value of one is used for q. The spatial derivatives of the function, φ,can then be achieved by solving Eq. (5), using the least-squares method [12]:

$$\left.\frac{\partial \phi}{\partial x}\right|_i = \sum_{j=1}^{m} a_{ij}\Delta \phi_{ij}, \qquad \left.\frac{\partial \phi}{\partial y}\right|_i = \sum_{j=1}^{m} b_{ij}\Delta \phi_{ij},$$
$$\left.\frac{\partial \phi}{\partial z}\right|_i = \sum_{j=1}^{n} c_{ij}\left(\phi_j - \phi_i\right). \tag{6}$$

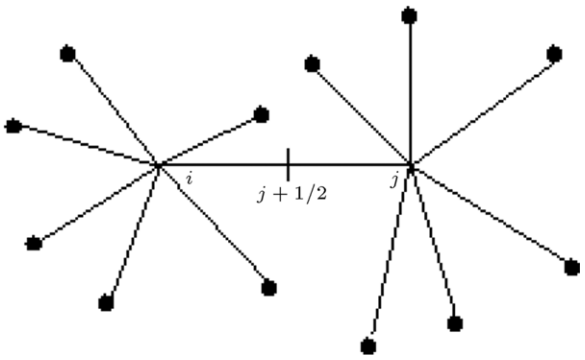

Figure 1: Schematic of point and its neighbors.

The coefficients in Eq. (6) can be calculated using the inverse distance weighting function as:

$$a_{ij} = \alpha_{ij,1} - \frac{r_{12}}{r_{11}}\alpha_{ij,2} + \beta\alpha_{ij,3},$$
$$b_{ij} = \alpha_{ij,2} - \frac{r_{23}}{r_{22}}\alpha_{ij,3}, \tag{7}$$
$$c_{ij} = \alpha_{ij,3},$$

where the parameters involved in the above equations are calculated as follows [12]:

$$r_{11} = \sqrt{\sum_{j=1}^{n} \omega_{ij}\left(\Delta x_{ij}\right)^2},$$
$$r_{12} = \frac{1}{r_{11}}\sum_{j=1}^{n} \omega_{ij}\Delta x_{ij}\Delta y_{ij},$$
$$r_{22} = \sqrt{\sum_{j=1}^{n} \omega_{ij}\left(\Delta y_{ij}\right)^2 - r_{12}^2},$$
$$r_{13} = \frac{1}{r_{11}}\sum_{j=1}^{n} \omega_{ij}\Delta x_{ij}\Delta z_{ij},$$
$$r_{23} = \frac{1}{r_{22}}\left(\sum_{j=1}^{n} \omega_{ij}\Delta y_{ij}\Delta z_{ij} - \frac{r_{12}}{r_{11}}\sum_{j=1}^{n} \omega_{ij}\Delta x_{ij}\Delta z_{ij}\right), \tag{8}$$
$$r_{33} = \sqrt{\sum_{j=1}^{n} \omega_{ij}\left(\Delta z_{ij}\right)^2 - \left(r_{13}^2 + r_{23}^2\right)},$$
$$\alpha_{ij,1} = \frac{\Delta x_{ij}}{r_{11}^2},$$
$$\alpha_{ij,2} = \frac{1}{r_{22}^2}\left(\Delta y_{ij} - \frac{r_{12}}{r_{11}}\Delta x_{ij}\right),$$
$$\alpha_{ij,3} = \frac{1}{r_{33}^2}\left(\Delta z_{ij} - \frac{r_{23}}{r_{22}}\Delta y_{ij} + \beta\Delta x_{ij}\right).$$

Applying the least-square approximations given by Eq. (6) to each component of the flux functions in Eq. (1), a semi-discrete form of the Euler equations at point i is obtained:

$$\left[\frac{\partial \mathbf{w}_i}{\partial t}\right] + \left[\sum_{j=1}^{m} a_{ij}\Delta F_{ij} + \sum_{j=1}^{m} b_{ij}\Delta G_{ij} + \sum_{j=1}^{m} c_{ij}\Delta K_{ij}\right] = 0. \tag{9}$$

Here, ΔF_{ij}, ΔG_{ij} and ΔK_{ij} are defined as:

$$\begin{aligned}\Delta F_{ij} &= F_j - F_i,\\ \Delta G_{ij} &= G_j - G_i,\\ \Delta K_{ij} &= K_j - K_i.\end{aligned} \tag{10}$$

Then, we define a Flux, $H = aF + bG + cK$, in the direction of the least-square coefficient vector for an edge, ij, similar to a directional flux towards a face area on an unstructured mesh. The approximation of Eq. (9) with the directed flux becomes:

$$\frac{\partial W_i}{\partial t} + \sum_{j=1}^{m} \Delta H_{ij} = 0. \tag{11}$$

The above equation demonstrates a non-dissipative unstable discretization. One may evaluate the flux, H, at the midpoint of edge (i, j) instead of point j. Thus, Eq. (11) is transformed into the following form:

$$\frac{\partial W_i}{\partial t} + 2\sum_{j=1}^{m} \Delta H_{ij+1/2} = 0. \tag{12}$$

A general idea for stabilizing the scheme is to add diffusive terms to Eq. (12). Two different approaches are investigated in this work; the CUSP scheme and scalar dissipation terms including a blend of second and fourth differences of flow variables.

2.1.1. CUSP scheme

For the sake of stabilizing Eq. (12), dissipation terms are added to the flux at the mid-point $(i, j + 1/2)$.

$$H_{j+1/2} = \frac{1}{2}\left(H_i + H_j\right) - \frac{1}{2}D_{j+1/2}. \tag{13}$$

The scheme capable of capturing shock with a single interior point may be shown as [6,7]:

$$D_{j+1/2} = \alpha^* c\,(W_R - W_L) + \beta\,(H(W_R) - H(W_L))\,. \tag{14}$$

Higher order accuracy is obtained by using SLIP reconstruction. In SLIP reconstruction, least squares gradients are used as follows [6,7]:

$$\begin{aligned}W_L &= W_i + \frac{1}{2}\Delta W, \qquad W_R = W_j - \frac{1}{2}\Delta W\\ \Delta W &= \frac{1}{2}R\left(\Delta W_i, \Delta W_j\right)\left(\Delta W_i + \Delta W_j\right)\\ \Delta W_i &= l_{ij}\cdot\nabla W_i, \qquad \Delta W_j = l_{ij}\cdot\nabla W_j.\end{aligned} \tag{15}$$

A limiter is used to suppress oscillations of the second and subsequent orders of solution in high gradient regions. The limiter that is used in this work is evaluated as:

$$R\left(\Delta W_i, \Delta W_j\right) = 1 - \left|\frac{\Delta W_i - \Delta W_j}{|\Delta W_i| + \left|\Delta W_j\right| + \varepsilon}\right|^r, \tag{16}$$

where ε is a very small number to put a stop to division by zero in smooth regions of the flow field. Whenever ΔW_i and ΔW_j have the opposite sign, as in the vicinity of a shock, the limiter, R, become zero and results in a first-order scheme for the artificial viscosity.

$$W_R - W_L = \Delta W_{ij}. \tag{17}$$

For smooth regions of the flow, $R\left(\Delta W_i, \Delta W_j\right) = 1$, and discretization behaves as a third difference. The scheme is not very sensitive to the value of the exponent, r, which is typically chosen to be 2 or 3. If one increases the values of r, it will result in less artificial dissipation. The switching functions ($\alpha^* c$ and β) are given by:

$$\alpha^* = \alpha - \beta\tilde{M}, \tag{18}$$

where c, $\tilde{M}$, α and β are evaluated as:

$$\begin{aligned}c &= \frac{(c_L + c_R)}{2}\\ c_L &= \sqrt{\gamma\frac{P_L}{\rho_L}}, \qquad c_R = \sqrt{\gamma\frac{P_R}{\rho_R}}\\ \tilde{M} &= \frac{\tilde{u}}{c} = \frac{(u_i + u_j)}{2 * c}\\ \alpha &= \begin{cases}\dfrac{1}{2}\left(\varepsilon + \dfrac{M^2}{\varepsilon}\right) & \text{if } |M| < \varepsilon\\ |M| & \text{if } |M| > \varepsilon\end{cases}\\ \beta &= \begin{cases}\text{Max}\,(0, 2M - 1) & \text{if } 0 < M < 1\\ \text{Min}\,(0, 2M + 1) & \text{if } -1 < M < 0\\ \dfrac{|M|}{M} & \text{if } |M| \geq 1\end{cases}\end{aligned} \tag{19}$$

c, the speed of sound at mid-point $(j + \frac{1}{2})$, is evaluated using Roe-averaged variables.

2.1.2. Scalar dissipation scheme

In the second dissipation model, an aggregation of the second and fourth differences of conserved variables (W) is added in order to prevent oscillations especially in the neighborhood of the shock waves.

$$\frac{\partial W_i}{\partial t} + 2\sum_{j=1}^{m} \Delta H_{ij+1/2} - D_i = 0. \tag{20}$$

These dissipation terms are defined by:

$$\begin{aligned}D_i &= \left(\nabla\left(\varepsilon^{(2)}\lambda\right)\nabla W - \nabla^2\left(\varepsilon^{(4)}\lambda\right)\nabla^2 W\right)_i\\ \nabla\left(\varepsilon^{(2)}\lambda\right)\nabla W &= \sum_{j=1}^{n}\left[\left(\varepsilon^{(2)}\lambda\right)_{i,j/2}\left(W_j - W_i\right)\right]\\ \nabla^2 W &= \sum_{j=1}^{n}\left(W_j - W_i\right)\end{aligned} \tag{21}$$

where $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are local adaptive coefficients that utilize the pressure as a sensor to explore sharp gradients. They are formulated as:

$$\begin{aligned}\varepsilon_{ij}^{(2)} &= k^{(2)}\nu_{ij}\\ \varepsilon_{ij}^{4} &= \max\left(0, k^{(4)} - \varepsilon_{ij}^{(2)}\right).\end{aligned} \tag{22}$$

Typical values of constant k_2 and k_4 are in the range $0 < k_2 < 1$ and $\frac{1}{256} < k_4 < \frac{1}{20}$ [13]. ν_{ij} is a pressure sensor for shocks at any ij edge that is described as:

$$\nu_{ij} = \frac{\left|P_j - P_i\right|}{\left|P_j + P_i\right|}. \tag{23}$$

λ_{ij} is defined as the largest eigenvalue in absolute, which equals the flux Jacobian matrix ($\frac{\partial F}{\partial W}$, $\frac{\partial G}{\partial W}$ and $\frac{\partial K}{\partial W}$) relevant to the Euler

equations.

$$\lambda_i = \sum_{j=1}^{m} \left(\left(a_{ij}\bar{u}_{j+1/2} + b_{ij}\bar{v}_{j+1/2} + c_{ij}\overline{w}_{j+1/2} \right) + \bar{c}_{j+1/2} L_{ij} \right)$$
$$L_{ij} = \sqrt{a_{ij}^2 + b_{ij}^2 + c_{ij}^2} \tag{24}$$

where $u_{ij+1/2}$, $v_{ij+1/2}$ and $w_{ij+1/2}$ are the velocity components and $c_{ij+1/2}$ is the speed of sound at the mid-point of ij [9].

2.2. Time discretization

Applying Eq. (20) to each node in the computational domain, the result will be a set of ordinary differential equations in the following form:

$$\frac{\partial W_i}{\partial t} + R(W_i) = 0$$
$$R(W_i) = 2\sum_{j=1}^{m} \Delta H_{ij+1/2} - D_i. \tag{25}$$

2.2.1. Explicit time integration

Eq. (25) in an explicit form can be presented as follows:

$$\frac{\partial W_i^{n+1}}{\partial t} + R\left(W_i^n\right) = 0. \tag{26}$$

The explicit four-stage Runge–Kutta scheme is used in this study.

$$W^{(0)} = W^{(n)}$$
$$W^{(1)} = W^{(0)} - \alpha_1 \Delta t R\left(w^{(0)}\right)$$
$$\cdots$$
$$W^{(4)} = W^{(0)} - \alpha_4 \Delta t R\left(w^{(3)}\right) \tag{27}$$
$$w^{(n+1)} = w^{(4)}$$
$$\alpha_1 = 0.333, \qquad \alpha_2 = 0.2667$$
$$\alpha_3 = 0.5, \qquad \alpha_4 = 1.0.$$

Local time stepping at any node may be calculated using nodes at its influence cloud:

$$\Delta t_i = \frac{\mathrm{CFL}}{\lambda_i}$$
$$\lambda_i = \sum_{j=1}^{m} \left(\left(a_{ij}\bar{u}_{j+1/2} + b_{ij}\bar{v}_{j+1/2} + c_{ij}\overline{w}_{j+1/2} \right) + \bar{c}_{j+1/2} L_{ij} \right) \tag{28}$$
$$L_{ij} = \sqrt{a_{ij}^2 + b_{ij}^2 + c_{ij}^2},$$

where $\tilde{u}$, $\tilde{v}$, $\tilde{w}$ and $\tilde{c}$ are the velocity components and speed of sound at mid-point, respectively. For the sake of improving computational efficiency, the dissipation function, D, is calculated merely at the first and third stages. These values are then applied for the subsequent second and fourth stages. To accelerate the convergence, residual smoothing and enthalpy damping are put into use in the present work [9].

2.2.2. Implicit time integration

The governing equations written in the discrete form of Eq. (19) are integrated in time by using a fully implicit time discretization [9] that leads to:

$$\frac{dW}{dt} + R\left(W_i^{n+1}\right) = 0, \tag{29}$$

where the superscript $(n+1)$ denotes the time level $(n+1)\Delta t$ of the approximation, and the $\frac{d}{dt}$ operator is approximated by an implicit backward difference formula of m order accuracy with the form:

$$\frac{d}{dt} = \frac{1}{\Delta t} = \sum_{q=1}^{m} \frac{1}{q} \left[\Delta^-\right]^q, \tag{30}$$

where:

$$\Delta^- W_i^{n+1} = W_i^{n+1} - W_i^n. \tag{31}$$

In the present work, a second order accurate time discretization is used. Thus, Eq. (27) turns to:

$$\frac{3W_i^{n+1}}{2\Delta t} - \frac{-2W_i^n}{\Delta t} + \frac{W_i^{n-1}}{2\Delta t} + R\left(W_i^{n+1}\right) = 0. \tag{32}$$

Eq. (32) for W_i^{n+1} is nonlinear and, therefore, cannot be solved in analytical fashion. At this stage, it is beneficial to redefine a new residual, R^*, referred to as an unsteady residual, which is equal to the left-hand side of Eq. (32).

$$R^*\left(W_i^{n+1}\right) = \frac{3W_i^{n+1}}{2\Delta t} - \frac{-2W_i^n}{\Delta t} + \frac{W_i^{n-1}}{2\Delta t} + R\left(W_i^{n+1}\right). \tag{33}$$

The new equation can be considered as the solution of a steady-state problem, which can then be solved with a time marching method by introducing a derivative, with respect to a fictitious pseudo-time, τ;

$$\frac{\partial W_i^{n+1}}{\partial \tau} + R^*\left(W_i^{n+1}\right) = 0. \tag{34}$$

The steady state solution to Eq. (34) satisfies:

$$\frac{\partial W_i^{n+1}}{\partial \tau} = 0, \tag{35}$$

which means it also satisfies $R^*\left(W_i^{n+1}\right) = 0$, and is consequently the solution of unsteady Eq. (33). The pseudo-time problem can be solved by applying any time-marching method designed to solve steady-state problems, utilizing any of the standard acceleration techniques. In this research, Eq. (34) is solved using the explicit Runge–Kutta multistage scheme. The time step for any inner explicit iteration (pseudo-time, τ) can be evaluated as

$$\tau_i = \min\left[\frac{\mathrm{CFL}_{\mathrm{explicit}}}{\sum_{j=1}^{n} \left(\left(a_{ij}\tilde{u} + b_{ij}\tilde{v} + c_{ij}\tilde{w} \right) + \tilde{c}\sqrt{a_{ij}^2 + b_{ij}^2 + c_{ij}^2} \right)}, \frac{2\Delta t_i}{3} \right], \tag{36}$$

where $\mathrm{CFL}_{\mathrm{exp}}$ is a courant number for inner explicit iterations. For calculations in this research, the real time step (CFL) of 100 000, and the pseudo-time step of about 2 are taken into consideration. For all test cases, the inner explicit solution iterates until the average of density residuals reaches the level of 0.01.

To accelerate convergence, local time (pseudo-time and real time) stepping and enthalpy damping are applied [14]. The original residuals, R^*, may be substituted by the smoothed residuals, $\bar{R}^*$, by solving the implicit equation, which yields:

$$\bar{R}_i^* = R_i^* + \varepsilon \nabla^2 \bar{R}_i^*. \tag{37}$$

At each point, i, $\nabla^2 \bar{R}_i^*$ represents the undivided Laplacian of the residuals, and ε is the smoothing coefficient, which is chosen as 0.5 in this research.

2.3. Boundary conditions

To solve Euler equations at a solid boundary, no mass or other convective fluxes can penetrate the solid body. The following five conditions replace the integrated solution, similar to boundary condition enforcement in a finite difference method [9]:

$$\frac{\partial P}{\partial n} = -\frac{\rho U_T^2}{R_s}, \qquad \frac{\partial \rho}{\partial n} = 0, \qquad \frac{\partial H}{\partial n} = 0, \qquad \frac{\partial u_t}{\partial n} = 0, \qquad u_n = 0, \tag{38}$$

where u_t is tangential velocity, u_n is normal velocity and H is total enthalpy. The four boundary conditions, which are conditions on derivatives, may be solved via the least-square framework, following the work of Hashemi and Jahangirian [9], except that ghost points are not used here because there are enough points in the neighboring field of boundary points.

In the far field, to determine the values of flow variables on the outer nodes, the characteristic analysis used is based on Riemann invariants. This analysis correctly accounts for wave propagations in the far field, which is a key factor for rapid convergence to steady state, and either serves as a non-reflecting boundary condition for unsteady applications. Riemann invariants, R, based on free stream and extrapolation values, are calculated as:

$$R_\infty = q_\infty \cdot n - \frac{2c_\infty}{(\gamma - 1)}, \qquad R_e = q_e \cdot n + \frac{2c_e}{(\gamma - 1)}. \tag{39}$$

The subscript, e, denotes values determined from extrapolated values, using Formulation (39), and the magnitude of normal velocity and the speed of sound at boundary nodes are found as:

$$q \cdot n = \frac{1}{2}(R_e + R_\infty), \qquad c = \frac{1}{4}(\gamma - 1)(R_e - R_\infty). \tag{40}$$

The above characteristic far field boundary conditions assume zero circulation, which is not correct for a lifting body. Consequently, the far field boundary has to be located reasonably far away from the body [10,14].

3. Results

The capabilities of the method are demonstrated by flow computations around different 3D geometries. The first test case is a rectangular wing, with NACA0012 section and a 5 chord span, at flow conditions of $M_\infty = 0.50, \alpha = 0^\circ$. In this case, the values of 0.45 and 0.02 are used for scalar dissipation coefficients, k_2 and k_4, respectively. The limiter function, r, is chosen to be 2, and a value of 0.0125 is used for ε to evaluate α in the CUSP dissipation scheme. It is noted that in all test cases, the CFL number, Runge–Kutta and residual smoothing coefficients are fixed [9]. The computations are executed on a Pentium PC Dual Core, 2.00 GHz speed.

In this case, the point distribution study has also been implemented using three computational point distributions, including 52 398, 159 244 and 207 341 points, regarded as coarse, medium and fine distributions, respectively. The medium point distribution over a NACA 0012 rectangular wing is shown in Figure 2. The far field is located about 8 chords away from the wing. There are 25 084 points on the solid boundary and 220 points on the outer boundary. The surface pressure coefficient distributions at the middle section of the wing, using different point distributions, are depicted in Figure 3. According to this figure, one can see that the coarse point results are not satisfactory, particularly near the leading edge of the wing section. However, no significant differences can be recognized between the medium and fine distribution results. Therefore, the medium distribution of points is chosen for the calculations.

The surface pressure coefficient distributions, using different dissipation models, are compared with the 2D control volume results in Figure 4. As shown, the results are in excellent

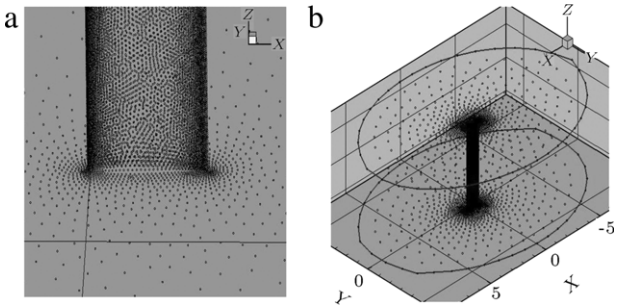

Figure 2: Point distribution for rectangular wing.

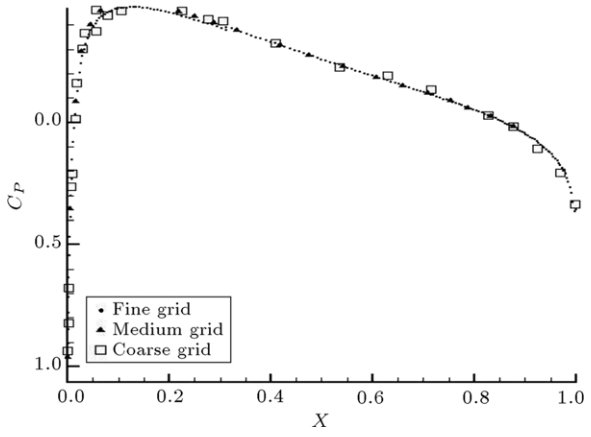

Figure 3: Surface pressure coefficient distributions at the middle section of the NACA0012 wing, $M_\infty = 0.5$, AOA $= 0^\circ$.

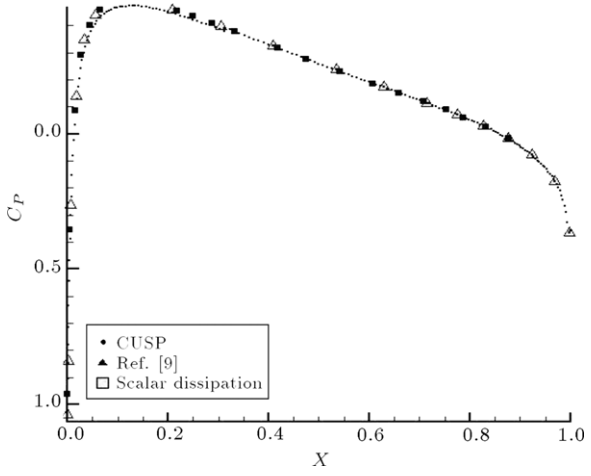

Figure 4: Surface pressure coefficient distributions at the middle section of the wing with different artificial dissipation techniques, $M_\infty = 0.5$, AOA $= 0.00^\circ$.

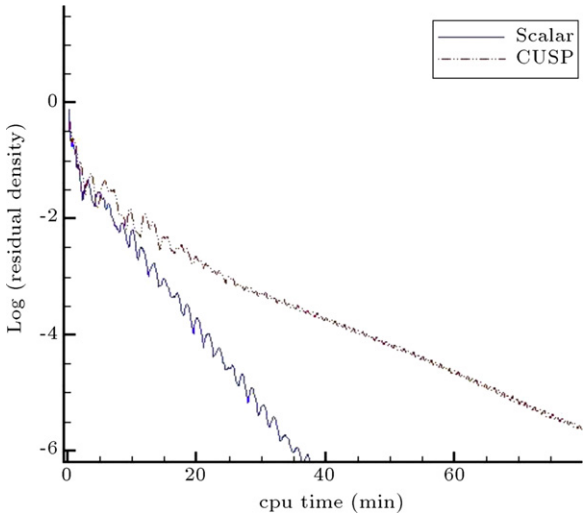

Figure 5: Convergence history for NACA 0012 wing at $M_\infty = 0.50, \alpha = 0^\circ$ (explicit).

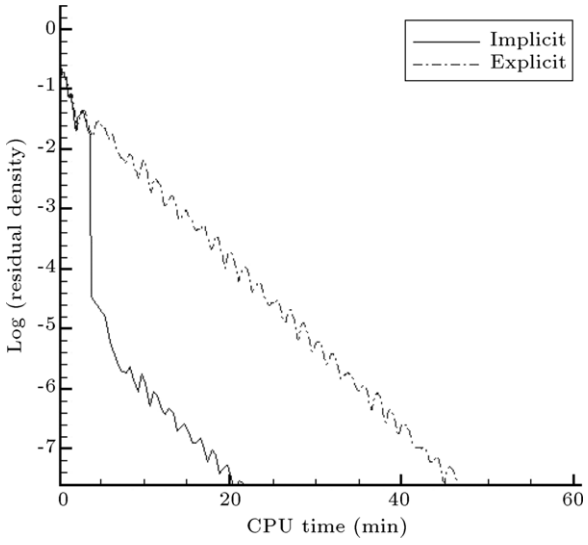

Figure 6: Convergence history for NACA 0012 wing at $M_\infty = 0.50, \alpha = 0^\circ$.

agreement with each other [12]. The convergence history for this case is shown in Figure 5, comparing the computational efficiency of different dissipation models. As illustrated, the scalar dissipation model that is presented in this paper takes 50% less computational time to reach a certain level of convergence. The computational efficiency provided by implicit and explicit counterpart methods are compared and illustrated in Figure 6. Note that the scalar dissipation model is used for both these methods in the mentioned figure. Again, more than 50% reduction in computational time is achieved when using the implicit scheme. Pressure contours in the middle section of the wing and around the surface of the wing are shown in Figure 7.

In order to demonstrate the ability of the method in the simulation of transonic flows with a relatively strong shock wave, the second case is chosen over the NACA 0012 wing at $M_\infty = 0.85, \alpha = 1.0^\circ$. In this case, the values of 0.6 and 0.02 are used for scalar dissipation coefficients, k_2 and k_4, respectively. The surface pressure coefficient distributions are compared in Figures 8 and 9. As illustrated, the results are in

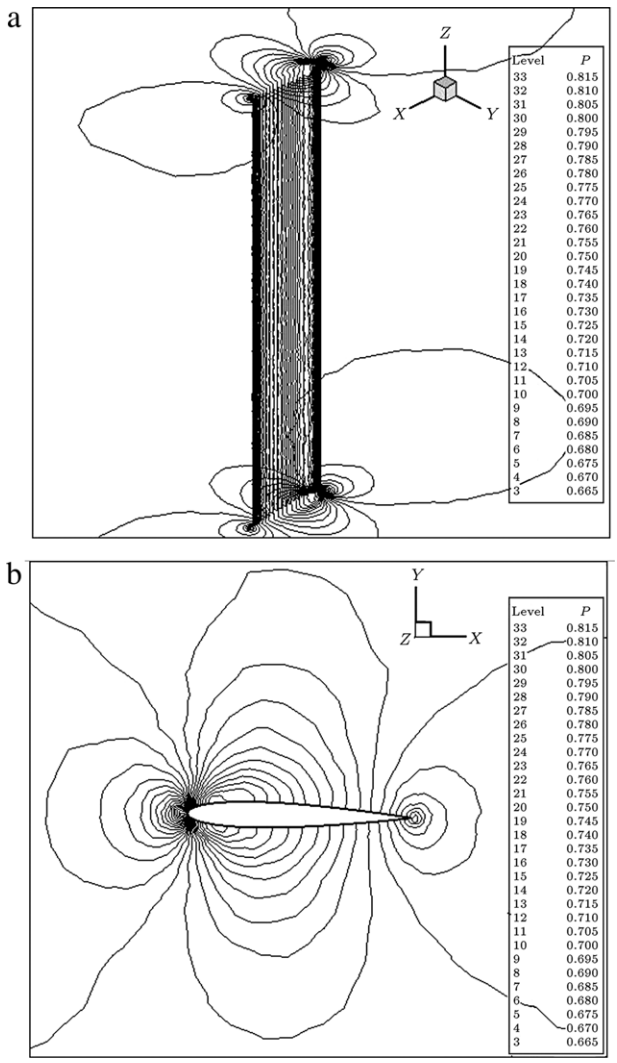

Figure 7: (a) Surface pressure contours and (b) pressure contours at middle section of the wing at $M_\infty = 0.5$ and AOA $= 0.00^\circ$.

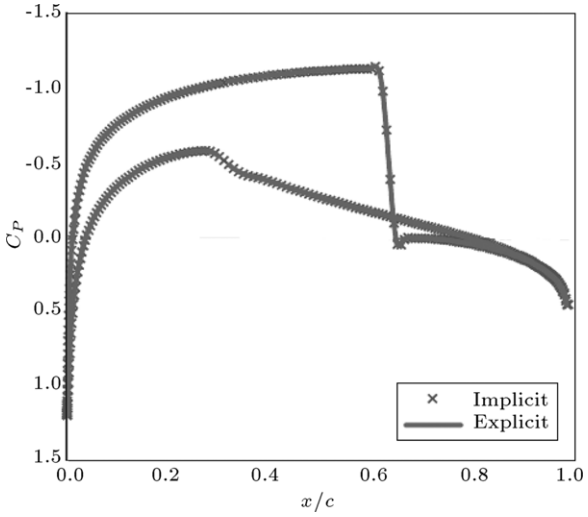

Figure 8: Surface pressure coefficient distributions in the middle section of the wing for implicit and explicit methods at $M_\infty = 0.85, \alpha = 1.0^\circ$.

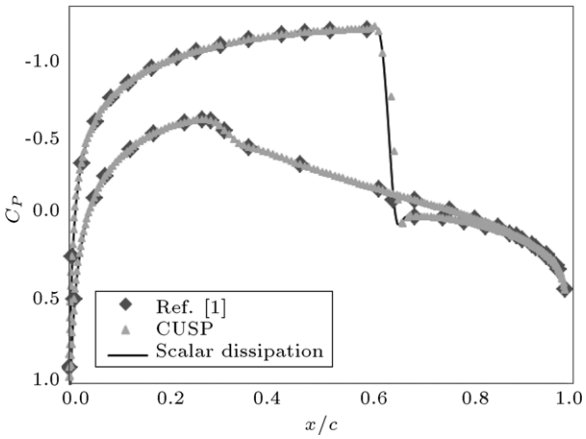

Figure 9: Surface pressure coefficient distributions in the middle section of the wing with different artificial dissipation techniques at $M_\infty = 0.85, \alpha = 1.0^\circ$.

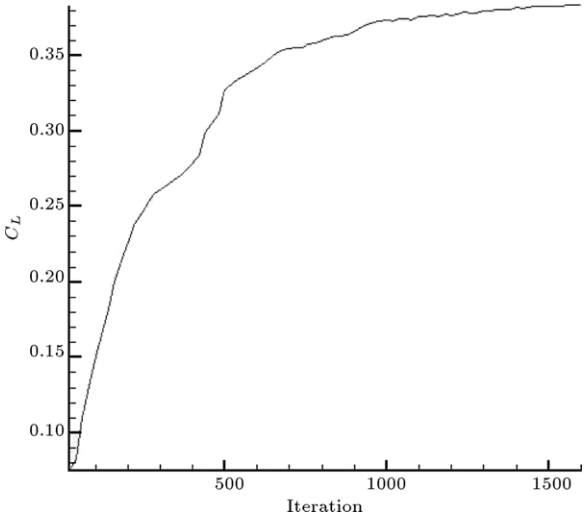

Figure 10: *cl* history for NACA 0012 wing at $M_\infty = 0.85, \alpha = 1.0^\circ$.

good agreement with each other, confirming the acceptable accuracy of the presented mesh-less method using the scalar dissipation model. The history of the calculated lift coefficient (C_L) is shown in Figure 10. The converged lift coefficient, in this case ($C_L = 0.3885$), has less than 1% difference with the numerical data reported by Katz and Jameson [6]. Furthermore, the convergence history of this case is shown in Figure 11, which proves the computational efficiency of the implicit mesh-less method over the alternative explicit approach. Figure 12 shows the pressure contours in the domain.

The third test case is defined to show the potential of the mesh-less method to simulate the flow over an ONERA M6 wing. The flow conditions chosen for this test case are $M_\infty = 0.8395, \alpha = 3.06^\circ$. The surface point distribution is shown in Figure 13. There are 145 496 points in the domain of which, 22 958 points are on the solid boundary. Surface pressure distributions at different sections of the wing, using the scalar dissipation model, are compared with experimental data and other methods [14] in Figures 14 and 15. Using the inviscid flow equations, it is expected to have stronger shock waves further downstream of its experimental position. However,

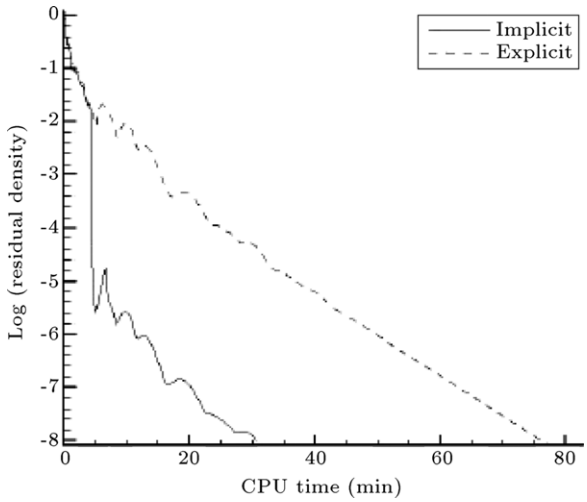

Figure 11: Convergence history for NACA 0012 wing at $M_\infty = 0.85, \alpha = 1.0^\circ$.

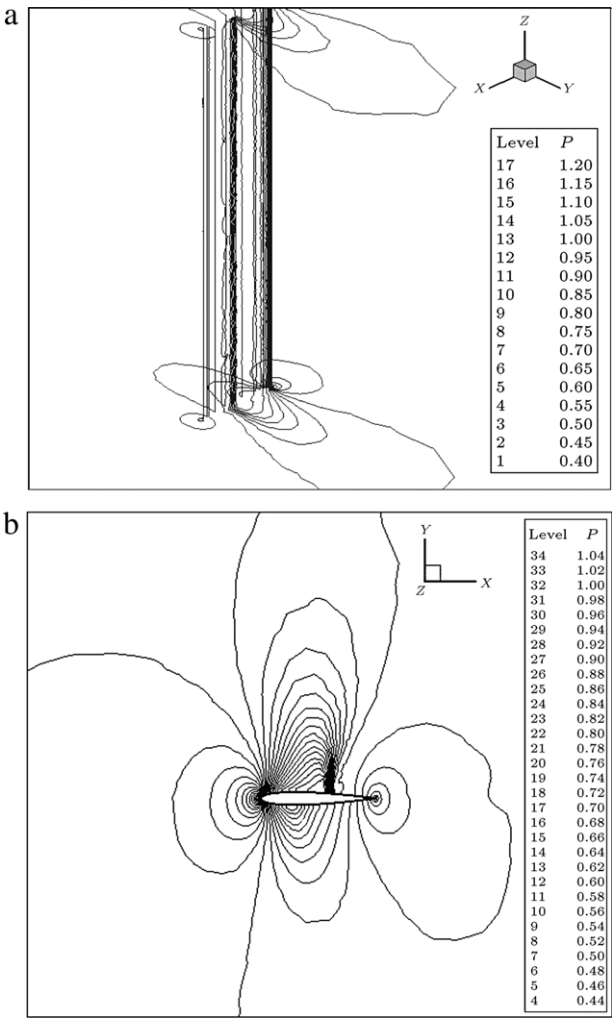

Figure 12: (a) Surface pressure contours, and (b) pressure contours at middle section of the wing at $M_\infty = 0.85, \alpha = 1.0^\circ$.

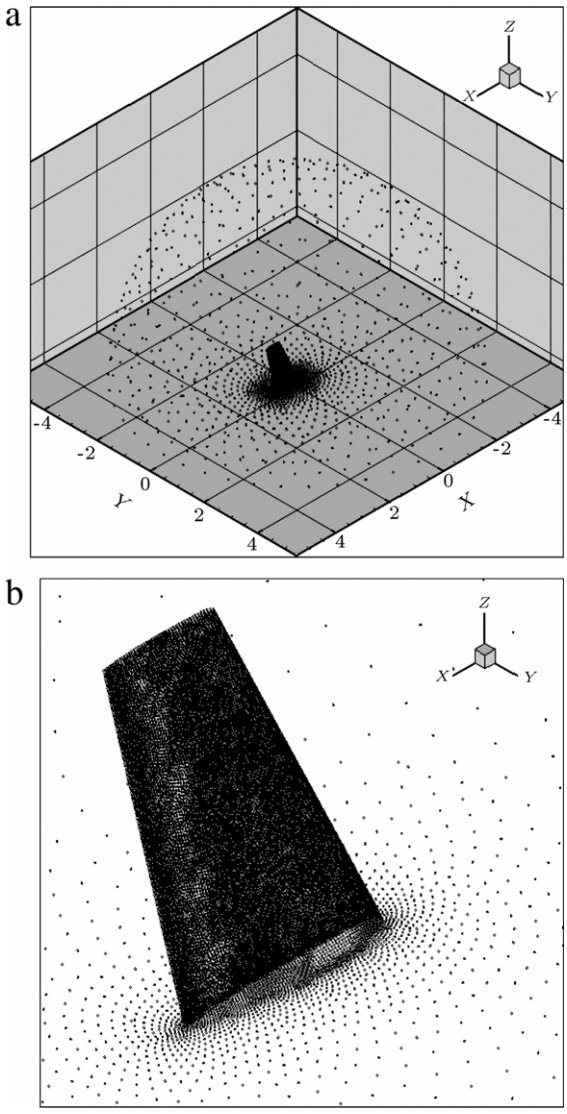

Figure 13: Boundary point distribution over ONERA M6 wing.

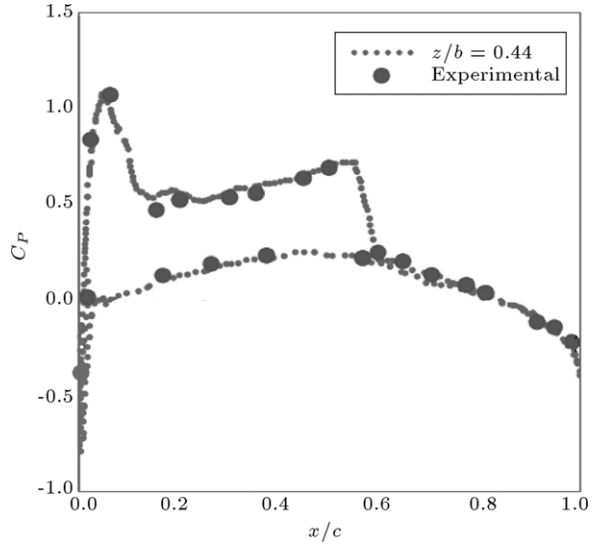

Figure 14: Pressure coefficient distributions over ONERA M6 wing in section $z/b = 0.44$ at $M_\infty = 0.8395$, $\alpha = 3.06^\circ$.

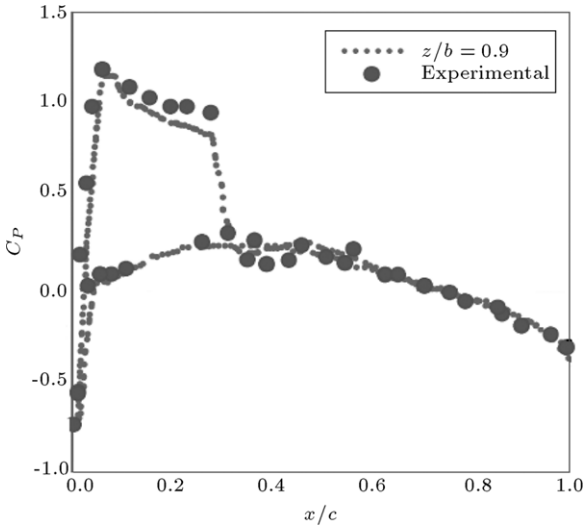

Figure 15: Pressure coefficient distributions over ONERA M6 wing in section $z/b = 0.9$ at $M_\infty = 0.8395$, $\alpha = 3.06^\circ$.

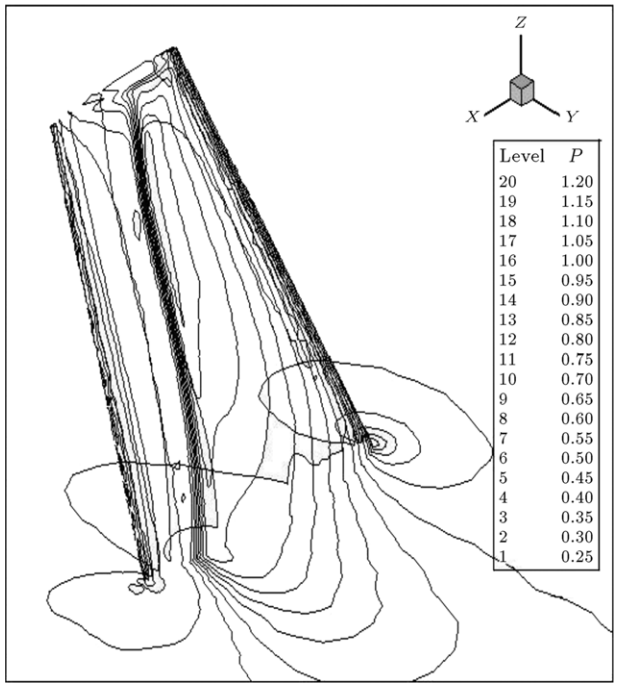

Figure 16: Pressure contours over ONERA M6 wing at $M_\infty = 0.8395$, $\alpha = 3.06^\circ$.

the results seem over-diffusive. This could be due to a relatively coarse distribution of points around the shock position within the domain. The surface pressure contours are demonstrated in Figure 16, showing the foot of the shock wave moving from the back of the wing, at the root sections, to the forward locations at the tip sections of the wing. The convergence history is also shown for this case, in comparison with the CUSP dissipation model in Figure 17. Again, remarkable computational efficiency is achieved using the scalar dissipation model.

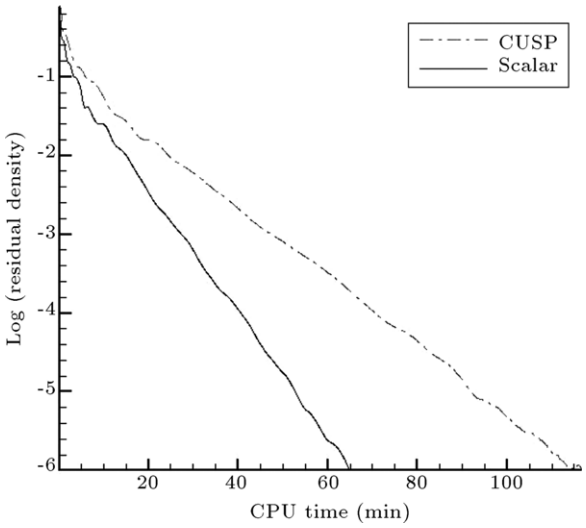

Figure 17: Convergence history for ONERA M6 wing.

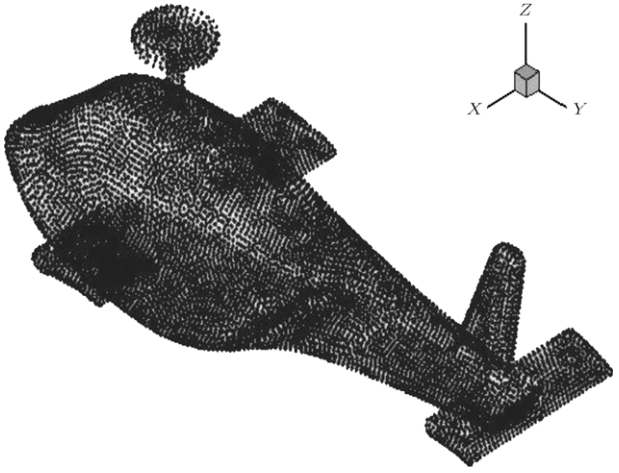

Figure 18: Point distribution over helicopter.

In order to show the applicability of the proposed method to more realistic geometries, a typical helicopter geometry is considered in the subsonic regime, i.e. $M = 0.3$. The point distribution on the surface geometry is shown in Figure 18 and the distribution of the points in a cross section around the helicopter is shown in Figure 19. There are 121 752 points in the domain, of which, 12 925 are on the surface. The Mach number contours around the body and surface pressure contours are shown in Figures 20 and 21, demonstrating the variations of flow properties for this application.

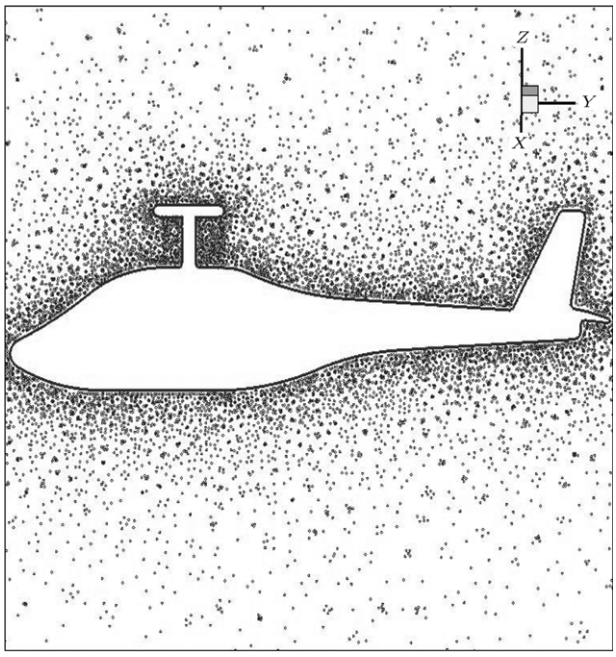

Figure 19: Point distribution around helicopter geometry.

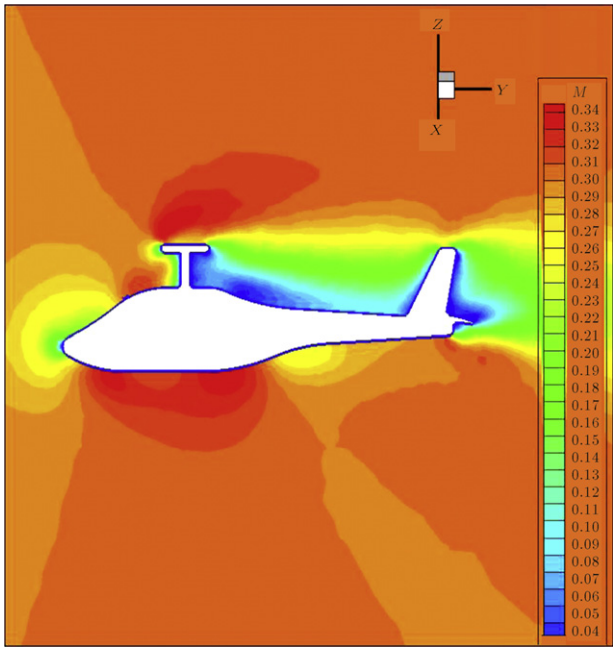

Figure 20: Mach number contours over helicopter, $M_\infty = 0.30$, by scalar dissipation model.

4. Conclusions

An implicit mesh-less method was presented for calculation of compressible flows around three dimensional complex geometries, and practical ones. The algorithm was applied directly to the differential form of the governing equations using least-square formulation. The computational efficiency was enhanced by adopting accelerating techniques, such as local time stepping and residual smoothing. Two artificial dissipation techniques were introduced for stability matters, and it was shown that the scalar one is more efficient, in terms of computational time, with the same level of accuracy. Results were presented which indicated good agreement with experimental and other reliable numerical data. The method was shown to reduce computational time by about 50% compared with the explicit method.

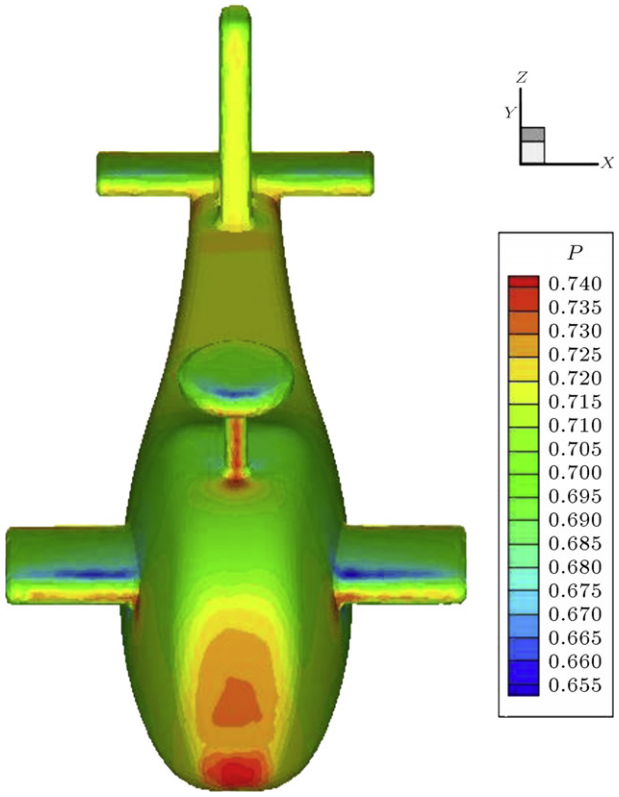

Figure 21: Pressure contours over helicopter at $M_\infty = 0.30$.

References

[1] Katz, A. and Jameson, A. "A comparison of various meshless schemes within a unified algorithm", *AIAA 2009–596, 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Orlando, Florida (January 5–8, 2009).

[2] Batina, J.T. "A gridless Euler/Navier–Stokes solution algorithm for complex 2-dimensional applications", NASA TM 107631 (1992).

[3] Onate, E., Idelsohn, S., Zienkiewicz, O.C., Taylor, R.L. and Sacco, C. "A stabilized finite point method for analysis of fluid mechanics problems", *Computer Methods in Applied Mechanics and Engineering*, 139, pp. 315–346 (1996).

[4] Deshpande, S.M., Anandhanarayanan, K., Praveen, C. and Ramesh, V. "Theory and application of 3D LSKUM based on entropy variables", *International Journal for Numerical Methods in Fluids*, 40, pp. 47–62 (2002).

[5] Srider, D. and Balakrishnan, N. "Convergence acceleration of an upwind least squares finite difference based meshless solver", *AIAA Journal*, 44, pp. 2189–2196 (2006).

[6] Katz, A. and Jameson, A. "Edge-based meshless methods for compressible flow simulations", *AIAA Paper 2008–699, 46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada (January 2008).

[7] Katz, A. and Jameson, A. "Edge-based meshless methods for compressible viscous flow with applications to overset grids", *AIAA Paper 2008–3989, AIAA 38th Fluid Dynamics Conference*, Seattle, WA (June 2008).

[8] Atluri, S.N., Kim, H.G. and Cho, J.Y. "A critical assessment of the truly mesh-less local Petrov–Galerkin (MLPG) and local boundary integral equation (LBIE) methods", *Computational Mechanics*, 24(5), pp. 348–372 (1999).

[9] Hashemi, M.Y. and Jahangirian, A. "An efficient implicit mesh-less method for compressible flow calculations", *International Journal for Numerical Methods in Fluids*, 67(6), pp. 754–770 (2011).

[10] Katz, A. and Jameson, A. "Multi-cloud: multi-grid convergence with a mesh-less operator", *Journal of Computational Physics*, 228(14), pp. 5237–5250 (2009).

[11] Jahangirian, A. and Hashemi, Y. "Hybrid unstructured Cartesian grid with mesh-less zones for compressible flow calculations", *11th ISGG Numerical Grid Generation Conference*, Montreal, Canada (May 25–28, 2009).

[12] Hashemi, M.Y. and Jahangirian, A. "Implicit fully mesh-less method for compressible viscous flow calculations", *Journal of Computational and Applied Mathematics*, 235(16), pp. 4687–4700 (2011).

[13] Jahangirian, A. and Hadidoolabi, M. "Unstructured moving grids for implicit calculation of unsteady compressible viscous flows", *International Journal for Numerical Methods in Fluids*, 47, pp. 1107–1113 (2005).

[14] Schmitt, V. and Charpin, F. "Pressure distributions on the ONERA-M6 wing at transonic mach numbers", *Experimental Data Base for Computer Program Assessment*, Report of the Fluid Dynamics Panel, Working Group 04, AGARD AR 138 (May 1979).

Samad Sattarzadeh received a M.S. degree in Aerospace Engineering from Amirkabir University of Technology (AUT), Tehran, Iran, where he is, currently, a Ph.D. degree student. His research interests include mesh-less methods.

Alireza Jahangirian received a B.S. degree in Mechanical Engineering from Amirkabir University of Technology (AUT), Tehran, Iran, in 1988, his M.S. degree in Mechanical Engineering from Sharif University of Technology, Tehran, Iran, in 1992, and a Ph.D. degree from Manchester University, England, in 1997. In 1998, he joined the Department of Aerospace Engineering, AUT, where he is currently Associate Professor. His research interests include: computational fluid dynamics, grid generation and evolutionary aerodynamic optimization.